AMENDMENTS TO THE SPECIFICATION

Please replace paragraph [0001] with the following amended paragraph:

[0001] This application is related to (1) U.S. Patent Application No. 09/996,088 entitled "AGGRESSIVE PREFETCH OF ADDRESS CHAINS," naming Peter Damron and Nicolai Kosche as inventors, and filed 28 November 2001 and to (2) U.S. Patent Application No. 10/050,387 ~~xx/xxx,xxx [Att'y Dkt. No. 004-7051]~~ entitled "TECHNIQUE FOR ASSOCIATING EXECUTION CHARACTERISTICS WITH INSTRUCTIONS OR OPERATIONS OF PROGRAM CODE," naming Nicolai Kosche, Christopher P. Aoki and Peter C. Damron as inventors, filed on even date herewith. Each of the related applications is incorporated herein by reference in its entirety.

Please replace paragraph [1015] with the following amended paragraph:

[1015] FIG. 6 is a flow chart illustrating preparation, in accordance with some embodiments of the present invention, of an instruction sequence that includes an unambiguous skid region to support/facilitate profiling.

Please replace paragraph [1020] with the following amended paragraph:

[1020] FIG. 1 depicts functional units of an illustrative processor 100 in which pipeline depth may contribute to delayed detection of execution events such as cache misses. As is illustrated, the processor 100 includes a floating point unit (FPU) 105 and an integer execution unit (IEU) 106. Profile-directed compilation techniques may be employed to prepare and/or optimize code for execution on processor 100 and, in some embodiments in accordance with the present invention, backtracking techniques may be employed to associate such execution events (or aggregations thereof) with particular instructions of the code and thereby guide code optimizations. For example, processor 100 includes a memory hierarchy for which latencies of some memory access instructions may be at least partially hidden using judicious placement of prefetch instructions as long as likely cache misses or other likely to stall conditions can be identified. Techniques in accordance with the present invention are particularly useful for the associating of delayed detections of cache misses with particular instructions so that cache miss likelihoods can be estimated.

Please replace paragraph [1021] with the following amended paragraph:

[1021] The memory hierarchy of processor 100 includes an on-board data cache 101 associated with a load/store unit 110 of the processor as well as a next level cache 102, 102A, main memory 104 and any intervening levels 103 (not specifically shown) of additional cache or buffering. As is shown, a memory interface unit 104A couples the main memory 104 to the cache 102. Persons of ordinary skill in the art will appreciate that in such hierarchies, latencies for memory accesses serviced from main memory rather than from cache, can be substantial. Accordingly, the payoff for reliably estimating cache miss likelihoods and, where possible hiding memory access latency, can be significant. While any of a variety of optimizations may benefit from techniques of the present invention, prefetch optimizations are illustrative. In this regard, co-pending U.S. Patent Application No. 09/996,088, entitled "AGGRESSIVE PREFETCH OF ADDRESS CHAINS," naming Peter C. Damron and Nicolai Kosche as inventors and filed 28 November 2001, the entirety of which is incorporated herein by reference, describes illustrative prefetch techniques that may benefit from techniques of the present invention that facilitate the association of instructions or operations with execution events, even in the presence of detection latencies. In particular, the above-incorporated patent application describes prefetch optimizations that exploit memory access latencies of "martyr operations." Candidate martyr operations, including likely-to-miss cache memory access instructions may be identified using techniques in accordance with the present invention.

Please replace paragraph [1031] with the following amended paragraph:

[1031] In either case, profile data is accumulated for associable target instructions. Typically, profile data is aggregated to provide a statistically valid characterization of individual target instructions based on criteria corresponding to the detected execution event. For example, a particular instruction for which collection indicates a normalized cache miss rate above a predetermined value ~~maybe~~ may be deemed to be a "likely cache miss." Other suitable characterizations are analogous.

Please replace the Abstract with the following amended paragraph:

Program code executed in an environment in which latency exists between an execution event and detection of the execution event may be profiled using a technique that includes backtracking from a point in a representation of the program code, which coincides with the detection toward a preceding operation associated with the execution event. Backtracking identifies the preceding operation at a displacement from the detection point unless an ambiguity creating location is disposed between the detection point and the preceding operation. In general, the relevant set of ambiguity creating locations is processor implementation dependent and program code specific; however, branch targets locations, entry points, and trap or interrupt handler locations are common examples. ~~In some realizations, the techniques may be used to associate cache miss (or hit) information with execution of particular memory access instructions. However, more generally, such techniques may be employed to associate observed execution characteristics with particular instructions of program code or associated operations based on event detections that may, in general, lag execution of the triggering instruction or operation by an interval that allows intervening program flow ambiguity.~~